ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA

Planteamiento y Validación de un Módulo de Aprendizaje para el IDS/IPS Snort

Abdón Carrera Rivera [1], Manuel Castillo Gutiérrez [2], Juan Quizhpi Ordóñez [3], Alfonso Aranda [4]
Facultad de Ingeniería en Electricidad y Computación (FIEC)
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil, Ecuador
abancarr@espol.edu.ec [1], mancasti@espol.edu.ec [2], cquizhpi@espol.edu.ec [3]
Escuela Superior Politécnica del Litoral (ESPOL) [4], Ingeniero en Computación [4], jaranda@espol.edu.ec [4]

Resumen

En este trabajo se presenta la implementación de un módulo de optimización para el IDS/IPS SNORT basando su mitigación no sólo en firmas, sino también mediante, estadísticas y aprendizaje.

El funcionamiento del módulo de optimización se basa en tres procesos que son modo aprendizaje, modo detección y limpieza de anomalías. El primero se centra en aprender los patrones del tráfico común, almacenándolo en una base de datos, luego se inicia el modo detección para comparar el tráfico aprendido previamente y detectar cualquier anomalía, enviando las conexiones anómalas al módulo de limpieza, lo cual permite a este último proceso eliminar única y exclusivamente la conexión maliciosa.

Palabras Clave: *Aprendizaje, Firmas, Anomalías, Snort.*

Abstract

This paper presents the implementation of an optimization module for IDS / IPS SNORT basing it's mitigation not only on signatures but also through statistics and learning methodes.

The operation of this module is based on three processes that are: learning mode, detection mode and anomalies clean up. The first process focuses on learning the common traffic patterns, stored those patterns in a database, then started the detection mode to compare the previously learned traffic and detect any abnormalities, finally abnormal and anomalous connections are sending the cleaning module, which allows this last process exclusively remove the malicious connection.

Key Words: *Learning, Signatures, Anomalies, Snort.*

1. Introducción

Los IDS/IPS son un avance en la seguridad perimetral, permiten no solo monitorear, sino también ejercer un control de acceso en una red informática, protegiéndola de accesos irregulares.

Debido a las grandes ventajas que proporcionan estos sistemas de seguridad informática, se planteó mejorar el IDS/IPS Snort basado en la detección de anomalías, es decir posibles ataques basados en comportamientos fuera del tráfico normal de la red.

2. Planteamiento

2.1. Definición del Problema

Hoy en día, los IDS/IPS generalmente usan la detección basada en firmas, comparan patrones de ataques ya conocidos con los datos que analizan, siendo el principal problema que solo detectan ataques conocidos, lo que imposibilita la detección de nuevas intrusiones. El problema se incrementa con el crecimiento de nuevos ataques y actualmente es posible encontrar en internet información que podría hacer que un usuario poco experimentado obtenga conocimientos básicos para poder acceder de forma poco honesta a algún sistema o red.

2.2. Objetivos

El objetivo general es aprovechar las fortalezas y oportunidades que nos brinda el IDS/IPS Snort, optimizando su funcionamiento.

Los siguientes objetivos específicos en el contexto del objetivo principal se detallan a continuación:

El funcionamiento se divide en tres etapas principales: Aprendizaje, Detección de anomalías y Limpieza de anomalías. En el aprendizaje, Snort capta y guarda todo el tráfico de una red, aprendiendo de esta. La detección de anomalías se apoya en un proceso donde se compara el tráfico normal que Snort aprendió de una red, alertando de posibles anomalías. Las detecciones maliciosas pasan al módulo de limpieza para poder evitar posibles ataques.

Desarrollar una interfaz gráfica de administración y monitoreo para mostrar información importante que respalde la toma de decisiones en tiempo oportuno.

3. Marco Teórico

3.1. Snort

Snort es un Sistema de Detección de Intrusiones basado en red y de código abierto. Además de realizar el análisis del tráfico en tiempo real, tiene un módulo con capacidad de generar alertas en tiempo real, registrando dichas alertas en ficheros de texto ASCII, UNIX sockets, bases de datos. (1)

Esquema de Snort

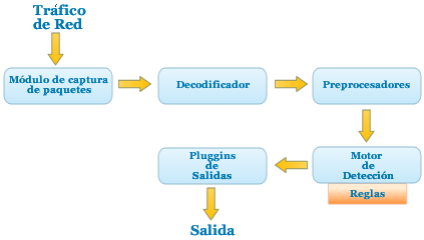

Figura 1. Esquema de Snort

Módulo de Captura de Paquetes.
Se encarga de la captura de paquetes.

Decodificador.
Decodifica los paquetes.

Preprocesadores.
El pre procesamiento permite a Snort analizar posteriormente los paquetes más fácilmente.

Motor de Detección.
Analiza paquetes de acuerdo a reglas predefinidas.

Pluggins de Salidas.
Las alertas pueden tener diversas formas de salida como logins, bases de datos y syslogs. (2)

3.2. Sistemas de Detección de Intrusos / Sistemas de Prevención de Intrusos (IDS/IPS)

Un IDS (Intrusion Detection System) es un sistema que intenta detectar o monitorizar eventos dentro de un equipo o a una red en busca de accesos no autorizados.

Un IPS (Intrusion Prevention Systems) es un sistema que permite establecer políticas de seguridad para proteger el equipo o la red de un ataque.

Se puede decir que un IPS brinda una protección proactiva y un IDS ofrece una protección reactiva. (3)

3.3. IDS/IPS con Detección de Anomalías

Se especializan en buscar actividad sospechosa dentro de un sistema, pero apoyándose en una fase inicial de aprendizaje que construye un perfil del sistema que se está monitoreando para luego mediante técnicas estadísticas comparar la información recibida en cada instante con el perfil creado. El aprendizaje se lleva a cabo durante un tiempo conveniente con el fin de considerar las actividades dentro del perfil como comportamiento normal y legítimo.

La Figura 2 muestra el esquema para elaborar un perfil

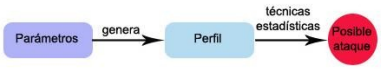

Figura 2. Creación de Perfil

Las medidas para la creación de un perfil podrían incluir una serie de parámetros como la carga de CPU, número de conexiones de red en una unidad de tiempo, número de procesos, entre otros. (4)

La Figura 3 resume las diferencias entre un IDS/IPS basados tanto en firmas como en anomalías.

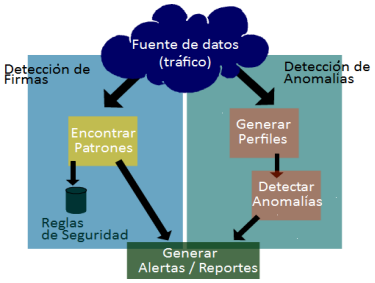

Figura 3. Diferencia IDS/IPS basado en firmas vs IDS/IPS basado en anomalías.

3.4. ACID

ACID (Analysis Console for Intrusion Databases) fue desarrollado por Roman Danyliw y es un motor de análisis basado en PHP que procesa una base de datos de incidentes de seguridad generados por IDS, firewalls y herramientas de monitoreo de red. Portátil, de código abierto y bajo licencia GPL, funciona en distintos sistemas operativos que soporten PHP como Linux y Windows. (5)

3.5. Firewall

Un Firewall es un sistema que impone políticas de seguridad entre una red privada y el Internet. Imposibilita el acceso de extraños a una computadora desde Internet. Pueden ser de dos tipos, de software o de hardware y proveen una frontera de protección contra intrusos. (6)

Iptables

Iptables es un sistema de firewall vinculado al kernel de Linux en espacio de usuario. Aplica políticas de filtrado del tráfico de una red. (7)

4. Análisis y Diseño

4.1. Análisis del Problema

Las debilidades más relevantes encontradas en el IDS/IPS Snort son:

1. No posee una base de conocimiento.
2. Solo realiza detecciones por medio de firmas.
3. No bloquea a los intrusos automáticamente.
4. No proporciona una interfaz gráfica.

Para el primer punto es muy importante poseer en una base de datos todo el tráfico de una red.

En el segundo punto se considera importante la detección por anomalías.

Para el tercer punto, son útiles ciertos preprocesadores para el bloqueo de conexiones.

En el último punto, una aplicación web para mostrar de una manera amigable los resultados de monitorear la red.

4.2. Diseño de la Solución

Al analizar el problema se determinó que existen varias etapas contenidas en la aplicación.

Monitoreo

El funcionamiento del IDS/IPS Snort es similar a los sniffers, porque el motor de análisis de Snort permite registrar y alertar ante un ataque previamente definido. La Figura 4 detalla esta etapa.

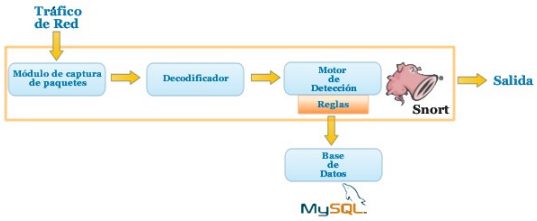

Figura 4. Etapa de Monitoreo y sus fases

Modo Aprendizaje

Antes de iniciar este modo, se necesita asignar la duración del tiempo en horas del aprendizaje del tráfico "normal" de una red. Cuando empieza el entrenamiento, se inicia también el IDS/IPS Snort en modo packet sniffer como un servicio más del sistema. Snort escanea el tráfico leyendo las cabeceras de los paquetes que pasen a través de la red y clasifica el tráfico dependiendo de su protocolo: TCP, UDP, ICMP, IP para luego almacenar todo lo escaneado en una base de datos. Se observa en la Figura 5 el contenido de cada perfil.

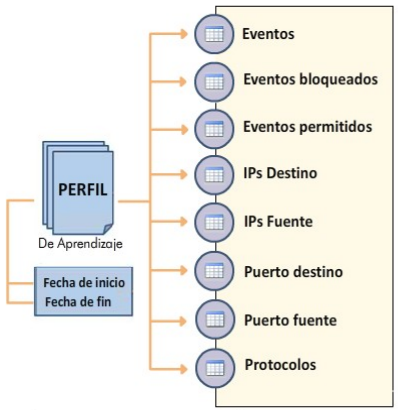

Figura 5. Partes del perfil de un aprendizaje

Luego, se procede a realizar el análisis estadístico creando histogramas y clasificando el tráfico por Ips Fuente, Ips Destino, Puertos fuente, Puertos destino y los eventos generales que ocurrieron en el tráfico.

Los histogramas son creados mientras se va entrenando el IDS/IPS Snort. El factor más importante del histograma es el número de conexiones, así como también el porcentaje de esa conexión en la red y el número de conexiones por minuto. Cada vez que cierta dirección IP realice una conexión, su valor total de ocurrencia va a aumentar. La Figura 6 muestra el proceso de aprendizaje.

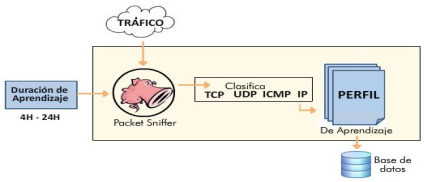

Figura 6. Modo Aprendizaje

Modo de Detección

Para iniciar este modo, primero se debe tener un perfil o conocimiento previo. En el modo detección también se guarda todo el tráfico que se escanea en la red y también se lo clasifica de la misma forma que en el modo aprendizaje, de esta manera se realizan comparaciones mediante los histogramas y umbrales.

Si al comparar las frecuencias de los histogramas obtenidos en la fase de aprendizaje con las frecuencias generadas en la fase de detección, se obtiene una desviación estándar mayor al umbral establecido por la frecuencia del aprendizaje, se considera algo anormal del tráfico común.

En base al ejemplo anterior, de lo obtenido en el modo aprendizaje se muestra los datos en la Tabla 1:

Ip Fuente	Puerto Fuente	Ip Destino	Puerto Destino	Protocolo	Ocurrencia
192.188.59.33	80	192.168.64.10	80	TCP	18

Umbral	Porcentaje en tráfico
0.05	5%

Tabla 1. Ejemplo del umbral de una dirección IP en el modo aprendizaje

Pero en el transcurso del modo detección obtuvimos la información presentada en la Tabla 2:

Ip Fuente	Puerto Fuente	Ip Destino	Puerto Destino	Protocolo	Ocurrencia
192.188.59.33	80	192.168.64.10	80	TCP	35

Umbral	Porcentaje en tráfico
0.097	10%

Tabla 2. Ejemplo del umbral de una dirección IP en el modo detección

Internamente se analizan los datos de las ocurrencias obtenidas en la detección y se los compara con los datos del entrenamiento, donde el valor de número de conexiones actuales (35) sobrepasa el umbral del tráfico común aprendido (18); casi duplicándolo, lo que quiere decir que se ha encontrado una anomalía. La Figura 7 muestra un histograma las ocurrencias de conexiones y su respectivo umbral

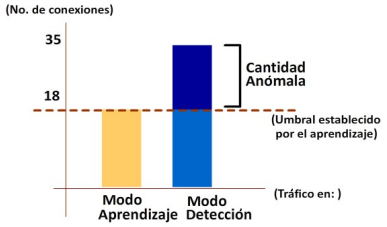

Figura 7. Histograma del tráfico en los modos aprendizaje y detección vs el número de conexiones

En tiempo real mientras se ejecuta la detección, se muestran las conexiones que violan la regla de sobrepasar el umbral de conexiones. Todas estas direcciones consideradas como anómalas, pasan a disparar alertas. La Figura 8 muestra las fases del módulo de detección

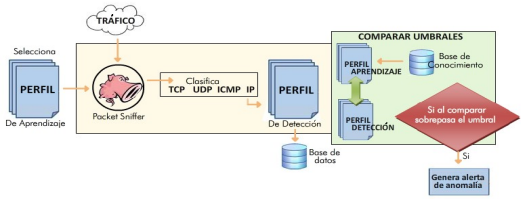

Figura 8. Módulo detección y sus fases

Limpieza de anomalías

Para eliminar o bloquear las conexiones anómalas se hace uso de iptables, para escribir los protocolos, direcciones fuente, destino, puertos fuente y destino que se desean bloquear. La Figura 9 representa cómo funciona el modo de limpieza

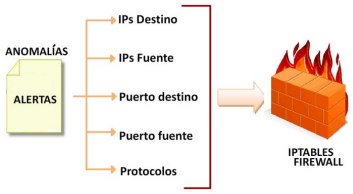

Figura 9. Modo de limpieza

Presentación

Para una mejor visualización de la información se implementó una aplicación web que permite un fácil acceso desde cualquier lugar y momento.

5. Implementación

5.1. Implementación Modo Aprendizaje

El modo aprendizaje al igual que todo el proyecto esta implementado en php y para una mejor comprensión del contenido de aprendizaje.php se lo ha resumido en tres secciones que se presentan a continuación:

Sección 1: Creación de Aprendizaje

Como se aprecia en la Figura 10, se escoge primero el tiempo de entrenamiento, la duración de este tiempo se lo selecciona de entre 4, 8, 16 y 24 horas.

Figura 10. Tiempo de duración de un aprendizaje

Después se da inicio al aprendizaje, se ejecuta el archivo createLearning.php que se encarga de crear la tabla learnings (Figura 11) dentro de la base de datos.

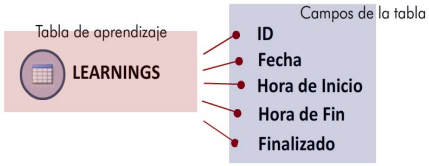

Figura 11. Estructura de la tabla learnings

El ID de cada perfil (Figura 12) es asociado con otras cinco tablas creadas.

Figura 12. Asociación de tablas con el aprendizaje

Donde ID es el identificador único de cada aprendizaje o perfil.

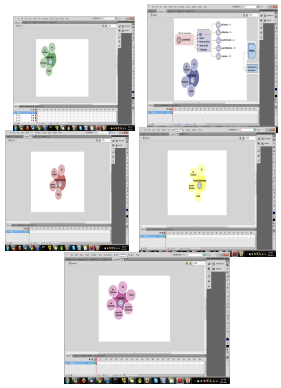

Figura 13. Tablas ipFuente, ipDestino, puertoFuente, puertoDestino y eventos

La Figura 13 muestra los campos de las cinco tablas, en las cuales se insertan o actualizan registros mediante un procedimiento almacenado que se ejecuta automáticamente antes de insertar algún registro en la tabla acid_event (contiene campos como ip fuente, ip destino, protocolo, puerto fuente, puerto destino) debido a que estos registros se los redistribuye a las tablas ya mencionadas.

Cada una de las cinco tablas posee un campo total que representa un contador que se inicializa o aumenta según se muestra a continuación con el ejemplo preliminar:

Para cada nuevo registro que pretende ser insertado en la tabla ipFuente27, previamente se compara en ipFuente y protocolo para determinar si existe otro registro con los mismos valores en esos campos, y de ser ese el caso se actualiza el registro ya contenido en la tabla aumentando en uno el campo total, caso contrario se inserta el nuevo registro y el campo total se inicializa en uno.

Así mismo, para las demás tablas del ejemplo se lleva de manera similar el proceso, pero con la diferencia en los campos de comparación que se presentan a continuación con su respectiva tabla:

Tabla ipDestino27, ipDestino y protocolo
Tabla puertoFuente27, puertoFuente e ipFuente
Tabla puertoDestino27, puertoDestino e ipDestino
Tabla eventos27, ipFuente, ipDestino, puertoFuente y puertoDestino

Por otro lado, Snort recoge los datos del tráfico del cual está aprendiendo mediante la ejecución del servicio snortstartAp:

```
exec('sudo service snortstartAp start',$salida);
```

La página se actualiza cada minuto mostrando los resultados

Sección 2: Detener Aprendizaje

También se presenta la opción de detener el aprendizaje, proceso que se establece mediante el archivo stopLearning.php cuyas funciones son finalizar el aprendizaje, actualizar el campo del correspondiente registro de la tabla learnings a finalizado=1, y eliminar el procedimiento almacenado creado inicialmente. La acción que se ejecuta para detener el servicio es la siguiente:

```
exec('sudo service snortstartAp stop',$salida);
```

Sección 3: Graficar Tablas

Muestra las cinco tablas de alertas del último aprendizaje e incluyen también los siguientes archivos que permiten graficar dichas tablas:

tablaEventos.php

Dependiendo del aprendizaje, se extrae de la base de datos los registros de la tabla eventos cuyas ip fuentes, puertos fuentes, ip destinos y puertos destinos no consten en las tablas eventosBloqueados y eventosPermitidos. Una vez obtenidos los registros se los incorpora a la interfaz para mostrarlos en la tabla Alertas de Eventos, que contiene los datos de:

IP Fuente, correspondiente al campo ipFuente de cada registro obtenido.

Puerto Fuente, correspondiente al campo puertoFuente de cada registro obtenido.

IP Destino, correspondiente al campo ipDestino de cada registro obtenido.

Puerto Destino, correspondiente al campo puertoDestino de cada registro obtenido.

De igual forma, las demás tablas de los archivos php tienen similitud, pero con diferentes campos en sus tablas, además de las tablas de bloqueados y permitidos como se observa a continuación:

tablaIpFuente.php

Campos: ipFuente y protocolo,
Tabla de extracción de registros: ipFuente27,
Otras tablas necesarias: ipFuenteBloqueados e ipFuentePermitidos

tablaIpDestino.php

Campos: ipDestino y protocolo,
Tabla de extracción de registros: ipDestino27,
Otras tablas necesarias: ipDestinoBloqueados e ipDestinoPermitidos

tablaPuertoFuente.php

Campos: puertoFuente e ipFuente,
Tabla de extracción de registros: puertoFuente27,
Otras tablas necesarias: puertoFuenteBloqueados y puertoFuentePermitidos

tablaPuertoDestino.php

Campos: puertoDestino e ipDestino,
Tabla de extracción de registros: puertoDestino27,
Otras tablas necesarias: puertoDestinoBloqueados y puertoDestinoPermitidos

Además, todas las tablas que se presentan en la interfaz, muestran los siguientes campos:

Ocurrencias en Aprendizaje, representa el número de conexiones de cada registro.

Ocurrencias en Aprendizaje %, es el porcentaje del número de conexiones de cada registro relativo al número de conexiones de todos los registros.

MNCM (Máximo Número de Conexiones por Minuto), se trata del número máximo de conexiones por minuto que puede alcanzar un registro.

La opción Control de Ip dentro del menú principal, restaura o bloquea las ips que han sido bloqueadas o permitidas respectivamente en el proceso.

5.2. Implementación Modo Detección

Para preparar la detección de anomalías de una red, primero se debe seleccionar un perfil de entrenamiento o aprendizaje creado previamente. Es por eso que se presentan todos los perfiles disponibles (Figura 14) con los cuales se va a comparar el tráfico común.

Figura 14. Perfiles disponibles para la detección

Una vez seleccionado el aprendizaje, se da inicio a la detección. Se realiza de manera similar a como se crea un aprendizaje. Dentro del código de createDetection,php se crea un perfil de detección (Figura 15), que compara el tráfico común y el aprendizaje de una manera más rápida y exacta. El único parámetro necesario para createDetection.php es el perfil base, que contiene el identificador ID del campo de la tabla learnings y el tipo de limpieza que se desea realizar en caso de surgir alertas de anomalías. De esta manera, con el ID del aprendizaje el sistema procede a comparar los tráficos de entrenamiento con los que recibe en tiempo real.

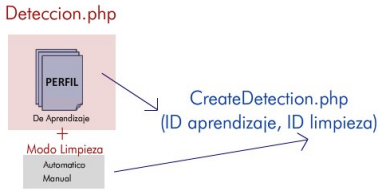

Figura 15. Parámetros para iniciar la detección

Se crea la tabla detección como se observa en la Figura 16

Figura 16. Estructura de la tabla detección

Los campos de la tabla detección son: id, fecha, hora de inicio, hora de fin, finalizado que determina si la detección sigue en ejecución, y el parámetro más importante, el ID, que se extrae desde que inicia el proceso de detección, y que tiene de referencia al aprendizaje de la base de conocimientos con el cual se compara, proveniente y almacenado en la tabla learnings de la base de Snort.

Una vez creada la tabla de detección (Figura 17), se procede a crear tablas dentro de la base de datos de snort, para almacenar el tráfico, donde se lo clasifica por ips fuente, ips destino, puertos fuente, puertos destino, protocolos y eventos generales.

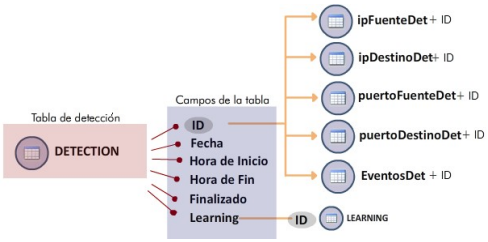

Figura 17. Estructura general y elementos que posee la detección

Luego, se efectúa el proceso de comparación que se lleva a cabo de acuerdo cada archivo php

tablaEventosDet.php

Por un lado se extrae la información de la tabla eventosDet que fue creado en el momento de iniciar la detección y por otro se extraen los registros de la tabla eventos del perfil escogido con el cual luego se realiza la comparación. Para ambos casos se determina un promedio como se detalla a continuación:

Promedio de detección, es el porcentaje del número de conexiones de un registro con respecto al número total de conexiones de todos los registros.

Promedio de aprendizaje, es la misma relación porcentual que el promedio de Detección con la diferencia de que los registros son extraídos de la tabla eventos.

La comparación se plantea con un índice de riesgo, que determina el peligro que alberga una conexión, y que inicialmente tiene un valor de cero.

Dentro del código, la variable del índice de riesgo tiene el nombre de $porcentajeC, la cual puede tomar diferentes valores de acuerdo a ciertas condiciones excluyentes que se explican a continuación:

Si el promedio de aprendizaje es mayor a cero y el promedio de detección es mayor al promedio de aprendizaje entonces el $porcentajeC = porcentajeC + 0.3;

Si el campo total del registro correspondiente a la tabla eventosDet es mayor al campo total del registro correspondiente a la tabla eventos entonces el $porcentajeC = porcentajeC + 0.3;

Si el número de conexiones por minuto del registro correspondiente a la tabla eventosDet es superior al número de conexiones por minuto del registro correspondiente a la tabla eventos entonces el $porcentajeC = porcentajeC + 0.4;

Los valores mencionados del promedio, del campo total y del campo mayor del aprendizaje representan umbrales que al ser superados por los correspondientes valores de detección producen que el índice de riesgo aumente y de acuerdo a ello se podrán bloquear manual o automáticamente ips y puertos respectivos.

El procedimiento anterior se generaliza para los cuatro restantes archivos php y sus correspondientes tablas de detección y de aprendizaje como se muestra:

tablaIpFuenteDet.php, tabla ipFuenteDet, tabla ipFuente

tablaIpDestinoDet.php, tabla ipDestinoDet, tabla ipDestino

tablaPuertoFuenteDet.php, tabla puertoFuenteDet, tabla puertoFuente

tablaPuertoDestinoDet.php, tabla puertoDestinoDet, tabla puertoDestino

5.3. Implementación Modo Limpieza

El proceso en el cual se bloquean o permiten ciertas conexiones, se denomina modo de limpieza.

Mientras se realiza el proceso de aprendizaje en tiempo real, se muestra toda la información de ips, puertos, protocolos o eventos generales con opciones de bloqueo como se observa en la Figura 18

Figura 18. Conexiones durante el aprendizaje con opciones de bloqueo o permitir

El bloqueo se realiza de manera opcional (Figura 19), debido a que durante el entrenamiento el administrador de la red desee omitir cualquier ip para que no sea contada durante el entrenamiento y el sistema no la tome en cuenta desde el inicio.

Figura 19. Tipo de Bloqueo

La primera alternativa es la limpieza automática, donde cualquier conexión que genere una alerta, se procede a bloquear automáticamente. Para la segunda opción, la limpieza es manual, donde un administrador de red puede observar una alerta disparada por una conexión anómala, y tomar una decisión.

Cuando una alerta se genera, ésta se presenta en su tabla correspondiente, mostrando toda su información. Si la alternativa seleccionada es manual, se presenta la opción de bloqueo de una conexión como se muestra en la Figura 20.

Figura 20. Opción de bloqueo durante el proceso de detección de anomalías

Cada botón hace referencia a una función de bloqueo o permisos dentro de las páginas permitirEvento.php y bloquearEvento.php respectivamente.

El archivo bloquearEvento.php se encarga de almacenar toda la información de la conexión a la base de datos para proceder a eliminar las conexiones y tener una constancia de cuales han sido las ips que generan anomalías, y si en algún momento se desea volver a permitir dicha conexión en la red el sistema presenta la opción de control de ips.

Los datos almacenados en la tabla eventosBloqueados de la base de datos snort y que se grafican en la Figura 21 son: ip fuente, ip destino, puerto fuente, puerto destino.

Figura 21. Estructura de la tabla eventos bloqueados

Una conexión anómala pasa a ser insertada en la tabla eventos bloqueados.

Luego se procede a bloquearla mediante el firewall con el uso de iptables, creando reglas y modificando la configuración del firewall.

Como parámetro se ejecuta la siguiente sentencia que escribe la regla para el firewall.

exec("sudo /sbin/iptables -I INPUT -s ".acidLong2IP ($_GET[ipFuente])." -p ".$proto." --sport ". $_GET[puertoFuente]." -d ".acidLong2IP($_GET[ipDestino])." --dport ". $_GET[puertoDestino]." -j DROP");

Donde la sentencia general está dada por:

Iptables –I INPUT -s IpFuente -p Protocolo --sport PuertoFuen –dport PuertoDest –j DROP

Se escribe la regla cambiando los valores de ipFuente, proto, puertoFuente, ipDestino, puertoDestino, por los valores de la conexión maliciosa almacenada en la base de datos de snort en la tabla eventosBloqueados.

El módulo de limpieza consta con una sección donde el administrador puede examinar todas las conexiones que han sido eliminadas por violar reglas de anomalías así como también se pueden observar las direcciones ip que tienen permisos y que no son contadas en la detección.

La administración de bloqueo se muestra en la Figura 22.

Figura 22. Control de direcciones ips bloqueadas o permitidas

6. Análisis de Resultados

6.1. Metodología de las pruebas

Para realizar las pruebas, se obtuvo de la empresa TELCONET por medio del departamento CERT un tráfico real de un cliente.

El tráfico solicitado ha sido obtenido en modo sniffer, realizando las pruebas los días 19 de Abril del 2011 y el 20 de Abril del 2011. En ambos días escaneando el tráfico desde las 4 PM hasta las 5 PM al cliente con un ancho de banda de 512kbits.

6.1. Resultados

Una vez concluido el primer proceso de entrenamiento en el primer día, se recolectaron los datos y se procedió a realizar los gráficos e inferir resultados a partir de los mismos.

Al finalizar el aprendizaje en la Tabla 3 se muestran el resultado de las 10 conexiones más concurridas.

Alertas de Eventos

IP Fuente	Puerto Fuente	IP Destino	Puerto Destino	Ocurrencias en Aprendizaje	Ocurrencias en Aprendizaje (%)	MNCM
172.21.0.156	8193	172.21.0.157	2142	60041	35.23719	19256
172.21.0.154	8193	172.21.0.155	2142	53278	31.26808	28038
172.21.0.44	8193	172.21.0.45	2142	2872	1.68554	392
172.21.0.52	8225	172.21.0.51	2142	2196	1.28880	396
172.21.0.50	66	172.21.0.51	2142	2190	1.28528	402
172.21.0.50	2	172.21.0.51	2142	2188	1.28411	400
172.21.0.130	8193	172.21.0.131	2142	1886	1.10687	312
172.21.0.130	8225	172.21.0.131	2142	1866	1.09513	310
10.11.16.177	646	224.0.0.2	646	1032	0.60567	80
192.168.11.1	646	224.0.0.2	646	1032	0.60567	78

Tabla 3. Resultado de 10 conexiones de mayor ocurrencia en aprendizaje

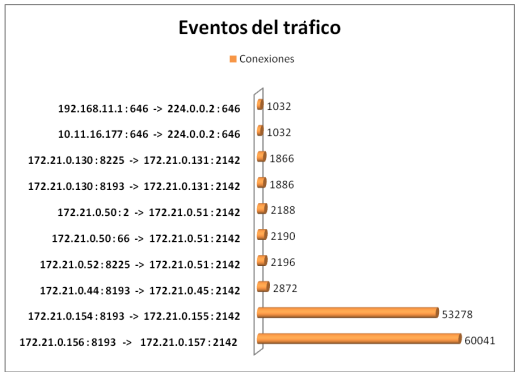

Figura 23. Conexiones sobresalientes del tráfico

En la Figura 23 se puede observar que la dirección fuente 172.21.0.156 : 8194 hacia la dirección destino 172.21.0.157 : 2142 posee el mayor número de ocurrencias en la red, con un valor de 60041 es la mayor cantidad aprendida por snort. Cabe recalcar el exceso de esta conexión al ser uno de los nodos de máximo uso del cliente.

Aproximadamente unas 1800 conexiones distintas, ya sean estas variaciones por puertos e IPs. El número de paquetes aprendidos se eleva aproximadamente a 1'233.775.

Con esta información se construye un umbral de referencia para el proceso posterior de detección.

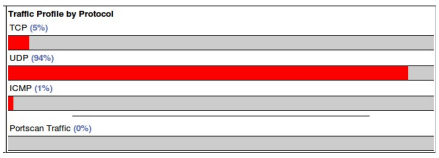

Figura 24. Tráfico escaneado en modo detección

S puede observar el perfil del tráfico general (Figura 24), dividido en un 5% TCP, 94% UDP y un mínimo de 1% para el tráfico ICMP.

Bajo las mismas condiciones del tráfico aprendido, se procedió el siguiente día a realizar la misma prueba, con la diferencia de que se buscó anomalías en la red.

Alertas de Eventos

IP Fuente	Puerto Fuente	IP Destino	Puerto Destino	Ocurrencias en Aprendizaje	Ocurrencias en Detección	Ocurrencias en Aprendizaje (%)	Ocurrencias en Detección (%)	MNCM en Aprendizaje	MNCM en Detección	IR
192.168.52.1	55597	172.21.0.156	3702	57	214	0.60567	1.10687	46	93	1
192.168.23.3		190.95.190.148		38	80	0.00117	0.01740	28	78	1
192.168.52.1	60309	239.255.255.250	1900	120	96	0.07043	0.41750	12	24	0.7
192.168.52.1	138	192.168.52.255	138	72	42	0.04226	0.18266	2	10	0.7
172.26.2.211	138	172.26.2.255	138	18	10	0.01056	0.04349	2	6	0.7

Tabla 4. Resultado de las anomalías detectadas

La tabla 4 muestra las anomalías registradas en el tráfico acorde con lo que Snort aprendió un día anterior.

Con un índice de riesgo (IR) igual a 1 se muestran las anomalías más destacadas e importantes las cuales se procedieron a eliminar por ser más graves, así también se tiene anomalías con un índice de riesgo de 0.7 de gran importancia.

En las siguientes figuras se puede apreciar porque la dirección fuente 192.168.52.1 genera una alerta de anomalía hacia la dirección ip 172.21.0.156

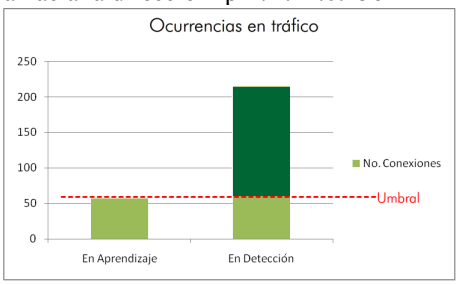

Figura 25. Comparación de umbrales en ocurrencias de tráfico

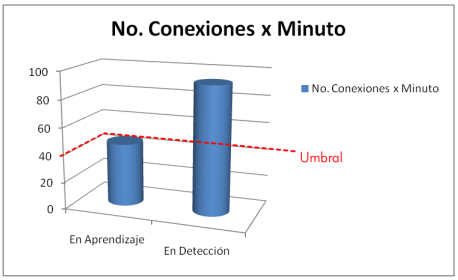

Figura 26. Comparación de umbrales en No. conexiones por minuto de tráfico

En ambos casos la conexión 192.168.52.1 : 55597 → 172.21.0.156 : 3702 supera el umbral establecido por el entrenamiento, una vez que sobrepasa las condiciones de los umbrales de número de conexiones máximas, porcentaje en el tráfico de la red y el mayor número de conexiones por minuto, se puede afirmar que dicha conexión es anómala con un índice de riesgo de 1.

7. Conclusiones y Recomendaciones

7.1. Conclusiones

1. Esta implementación basada en comportamientos permite sólo una protección adicional a la red.
2. Utilizar este módulo no asegura que una alerta generada sea efectivamente un ataque a la red, puede que se genere un falso positivo.
3. Establecer correctamente los factores para formar el índice de riesgo reduce el número de falsos positivos.
4. Otra forma de reducir falsos positivos es realizar el aprendizaje en momentos en que la red se encuentre en su mayor uso.
5. El uso de este módulo no reducirá a 0% el riesgo de ataques en la red.

7.2. Recomendaciones

1. Establecer políticas de seguridad con respecto al comportamiento de la red contribuye a una mejor detección de posibles ataques.
2. Aumentar los factores para el índice de riesgo y así reducir falsos positivos.
3. E recomendable eliminar el tráfico en desuso que se encuentra en la base de datos.
4. Entrenar de 8 a 24 horas el IDS/IPS Snort para que adquiera un mejor conocimiento del tráfico,.
5. El aprendizaje puede tener sus falencias, debido a que el tráfico con el cual se entrena, haciendo que tráfico anómalo pase como normal.
6. Existen otras formas para la etapa de entrenamiento como por ejemplo las redes neuronales, las cuales representarían una aproximación más real al comportamiento de un tráfico.

7. Bibliografía

[1] Armando Mira. "Tutorial de Snort". http://club.telepolis.com/websecure/tutoriales/tutorial_snort.pdf [Online] [Cited 2005].

[2] Carlos Jiménez Galindo. "Proyecto de Fin De Carrera". http://www.adminso.es/wiki/images/d/d0/Pfc_Carlos_cap3.pdf [Online] [Cited 2007].

[3] Pablo Martínez. "Análisis de Snort". http://pmartinez.files.wordpress.com/2007/07/analisis_snort.pdf [Online] [Cited 2006].

[4] Diego González Gómez. "Sistema de Detección de Intrusiones". http://www.dgonzalez.net/pub/ids/trans/IDStCOL.pdf [Online] [Cited Julio 2003].

[5] Roman Danyliw. "Analysis Console for Intrusion Databases". http://www.andrew.cmu.edu/user/rdanyliw/snort/snortacid.html [Online] [Cited 2003].

[6] José Madrigal Garcia, Mateo Luna Luna. "Firewalls". http://www.monografias.com/trabajos14/firewalls/firewalls.shtml [Online] [Cited 2007].

[7] Enrique González Rodríguez, Diego Trujillo García. "Firewall, Iptables, Proxy". http://serdis.dis.ulpgc.es/~a013775/asignaturas/ii-aso/curso0607/trabajos/seguridad/filtros/filtros_t.pdf [Online] [Cited Febrero 2007].